Attorney Docket No.: 020510-004900US
Client Reference No.: 16348-108CA

PATENT APPLICATION

METHOD AND APPARATUS FOR DETECTING VOICE ACTIVITY

Inventor(s): Song Zhang, a citizen of Canada, residing at
32 Sherk Crescent
Kanata, Ontario, K2K 2L3 Canada

Eric Verreault, a citizen of Canada, residing at
12-39 Putnam Avenue
Ottawa, Ontario, K1M 1Z1 Canada

Assignee: Catena Networks, Inc.
303 Twin Dolphin Drive, Suite 600
Redwood Shores, CA 94065

Entity: Small business concern

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 650-326-2400

METHOD AND APPARATUS FOR DETECTING VOICE ACTIVITY

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application claims priority from Canadian Patent Application No. 2,420,129 filed February 17, 2003

STATEMENT AS TO RIGHTS TO INVENTIONS MADE UNDER FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] NOT APPLICABLE

REFERENCE TO A "SEQUENCE LISTING," A TABLE, OR A COMPUTER PROGRAM LISTING APPENDIX SUBMITTED ON A COMPACT DISK.

[0003] NOT APPLICABLE

BACKGROUND OF THE INVENTION

[0004] The present invention relates generally to signal processing and specifically to a method for processing a signal for detecting voice activity.

[0005] Voice activity detection (VAD) techniques have been widely used in digital voice communications to decide when to enable reduction of a voice data rate to achieve either spectral-efficient voice transmission or power-efficient voice transmission. Such savings are particularly beneficial for wireless and other devices where spectrum and power limitations are an important factor. An essential part of VAD algorithms is to effectively distinguish a voice signal from a background noise signal, where multiple aspects of signal characteristics such as energy level, spectral contents, periodicity, stationarity, and the like have to be explored.

[0006] Traditional VAD algorithms tend to use heuristic approaches to apply a limited subset of the characteristics to detect voice presence. In practice, it is difficult to achieve a high voice detection rate and low false detection rate due to the heuristic nature of these techniques.

[0007] To address the performance issue of heuristic algorithms, more sophisticated algorithms have been developed to simultaneously monitor multiple signal characteristics and try to make a detection decision based on joint metrics. These algorithms demonstrate good performance, but often lead to complicated implementations or, inevitably, become an integrated component of a specific voice encoder algorithm.

[0008] Lately, a statistical model based VAD algorithm has been studied and yields good performance and a simple mathematical framework. This algorithm is described in detail in "A Statistical Model-Based Voice Activity Detection", Jongseo Sohn, Nam Soo Kim, and Wonyong Sung, IEEE Signal Processing Letters, Vol. 6, No. 1, Jan. 1999. The challenge, however, lies in applying this new algorithm to effectively distinguish voice and noise signals, as assumptions or prior knowledge of the SNR is required.

[0009] Accordingly, it is an object of the present invention to obviate or mitigate at least some of the abovementioned disadvantages.

BRIEF SUMMARY OF THE INVENTION

[0010] In accordance with an aspect of the present invention, there is provided a method for voice activity detection on an input signal using a log likelihood ratio (LLR), comprising the steps of: determining and tracking the signal's instant, minimum and maximum power levels; selecting a first predefined range of signals to be considered as noise; selecting a second predefined range of signals to be considered as voice; using the voice, noise and power signals for calculating the LLR; using the LLR for determining a threshold; and using the threshold for differentiating between noise and voice.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] An embodiment of the present invention will now be described by way example only with reference to the following drawings in which:

[0012] Figure 1 is a flow diagram illustrating the operation of a VAD algorithm according to an embodiment of the present invention;

[0013] Figure 2 is a graph illustrating a sample noise corrupted voice signal;

[0014] Figure 3 is a graph illustrating signal dynamics of a sample noise corrupted voice signal;

[0015] Figure 4 is a graph illustrating the establishment and tracking of minimum and maximum signal levels;

[0016] Figure 5 is a graph illustrating the establishment of a noise power profile;

[0017] Figure 6 is a graph illustrating the establishment of a voice power profile;

[0018] Figure 7 is a graph illustrating the establishment and tracking of a pri-SNR profile;

[0019] Figure 8 is a graph illustrating the LLR distribution over time;

[0020] Figure 9 is an enlarged view of a portion of the graph in Figure 8;

[0021] Figure 10 is a graph illustrating a noise suppressed voice signal; and

[0022] Figure 11 is a block diagram of a communications device according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0023] For convenience, like numerals in the description refer to like structures in the drawings. The following describes a robust statistical model-based VAD algorithm. The algorithm does not rely on any presumptions of voice and noise statistical characters and can quickly train itself to effectively detect voice signal with good performance. Further, it works as a stand-alone module and is independent of the type of voice encoders implemented.

[0024] The method described herein provides several advantages, including the use of a statistical model based approach with proven performance and simplicity, and self-training and adapting without reliance on any presumptions of voice and noise statistical characters. The method provides an adaptive detection threshold that makes the algorithm work in a wide range of signal-to-noise ratio (SNR) scenarios, particularly low SNR applications with a low false detection rate, and a generic stand-alone structure that can work with different voice encoders.

[0025] The underlying mathematical framework for the algorithm is the log likelihood ratio (LLR) of the event when there is noise only, and of the event when there are both voice and noise. These events can be mathematically formulated as follows.

[0026] A frame of a received signal is defined as *y(t)*, where $y(t) = x(t) + n(t)$, and where *x(t)* is a voice signal and *n(t)* is a noise signal. A corresponding pre-selected set of complex frequency components of y(t) is defined as *Y*.

[0027] Further, two events are defined as H_0 and H_1. H_0 is the event where speech is absent and thus $Y = N$, where *N* is a corresponding pre-selected set of complex frequency components of the noise signal *n(t)*. H_1 is the event where speech is present and thus $Y = X + N$, where *X* is a corresponding pre-selected set of complex frequency components of the voice signal x*(t)*.

[0028] It is sufficiently accurate to model *Y* as a jointly Gaussian distributed random vector with each individual component as an independent complex Gaussian variable, and *Y*'s probability density function (PDF) conditioned on H_0 and H_1 can be expressed as:

$$p(Y \mid H_0) = \prod_{k=0}^{L-1} \frac{1}{\pi \lambda_N(k)} \exp\left(-\frac{|Y_k|^2}{\lambda_N(k)} \right)$$

$$p(Y \mid H_1) = \prod_{k=0}^{L-1} \frac{1}{\pi [\lambda_X(k) + \lambda_N(k)]} \exp\left(-\frac{|Y_k|^2}{[\lambda_X(k) + \lambda_N(k)]} \right)$$

where $\lambda_X(k)$ and $\lambda_N(k)$ are the variances of the voice complex frequency component X_k and the noise complex frequency component N_k, respectively.

[0029] The log likelihood ratio (LLR) of the k^{th} frequency component is defined as:

$$\log(\Lambda_k) = \log\left(\frac{p(Y_k \mid H_1)}{p(Y_k \mid H_0)}\right) = \left(\frac{\gamma_k \cdot \xi_k}{1 + \xi_k}\right) - \log(1 + \xi_k)$$

where, ξ_k and γ_k are the a priori signal-to-noise ratio (pri-SNR) and a posteriori signal-to-noise ratios (post-SNR) respectively, and are defined by:

$$\xi_k = \frac{\lambda_X(k)}{\lambda_N(k)} \qquad \text{Equation 1}$$

$$\gamma_k = \frac{|Y_k|^2}{\lambda_N(k)} \qquad \text{Equation 2}$$

[0030] Then, the LLR of vector Y given H_0 and H_1, which is what a VAD decision may be based on, can expressed as:

$$\log(\Lambda) = \sum_k \log(\Lambda_k) = \sum_k \log(\frac{p(Y_k \mid H_1)}{p(Y_k \mid H_0)}) = \sum_k (\left(\frac{\gamma_k \cdot \xi_k}{1+\xi_k}\right) - \log(1+\xi_k)) \qquad \text{Equation 3}$$

A LLR threshold can be developed based on SNR levels, and can be used to make a decision as to whether the voice signal is present or not.

[0031] Referring to Figure 1, a flow chart illustrating the operation of a VAD algorithm in accordance with an embodiment of the invention is shown generally by numeral 100. In step 102, over a given period of time, an inbound signal is transformed from the time domain to the frequency domain by a Fast Fourier Transform, and the signal power on each frequency component is calculated. In step 104, the sum of the signal power over a pre-selected frequency range is calculated. In step 106, the sum of the signal power is passed through a first order Infinite Impulse Response (IIR) averaging filter for extracting frame averaged dynamics of the signal power. In step 108, the envelope of the power dynamics is extracted and tracked to build a minimum and maximum power level. In step 110, using the minimum and maximum power level as a reference, two power ranges are established: a noise power range and a voice power range. For each frame whose power falls into either of the two ranges, its per frequency power components are used to calculate the frame averaged per frequency noise power or voice power respectively. In step 111, noise and voice powers are averaged once per frequency over multiple frames, and they are used to calculate the a priori signal-to-noise ratio (pri-SNR) per frequency in accordance with Equation 1. In step 112, a per frequency posteriori SNR (post-SNR) is calculated on per frame basis in accordance with Equation 2. In step 113, the post-SNR and the pri-SNR are used to calculate the per frame LLR value in accordance according with Equation 3. In step 114, a LLR threshold is determined for making a VAD decision. In step 116, as the LLR threshold becomes available, the algorithm enters into a normal operation mode, where each frame's LLR value is calculated in accordance with Equation 3. The VAD decision

for each frame is made by comparing the frame LLR value against established noise LLR threshold. In the meantime, the quantities established in steps 106, 108, 110, 111, 112 and 114 are updated on a frame by frame basis.

[0032] One way of implementing the operation of the VAD algorithm illustrated in Figure 1 is described in detail as follows. Referring to Figure 2, a sample input signal is illustrated. (See also line 150 in Figure 1.) The input signal represents a combination of voice and noise signals of varying amplitude over a period of time. Each inbound 5 ms signal frame comprises 40 samples. In step 102, for each frame, a 32 or 64-point FFT is performed. If a 32-point FFT is performed, the 40-sample frame is truncated to 32 samples. If a 64-point FFT is performed, the 40-sample frame is zero padded. It will be appreciated by a person skilled in the art that the inbound signal frame size and FFT size can vary in accordance with the implementation.

[0033] In step 104, the sum of signal power over the pre-selected frequency set is calculated from the FFT output. Typically, the frequency set is selected such that it sufficiently covers the voice signal's power. In step 106, the sum of signal power is filtered through a first-order IIR averaging filter for extracting the frame-averaged signal power dynamics. The IIR averaging filter's forgetting factor is selected such that signal power's peaks and valleys are maintained. Referring to Figure 3, a sample output signal of the IIR averaging filter is shown. (See also line 152 in Figure 1.) The output signal represents the power dynamic of the input signal over a number of frames

[0034] The next step 108 is to determine minimum and maximum power levels and to track these power levels as they progress. One way of determining the initial minimum and maximum signal levels is described as follows. Since the signal's power dynamic is available from the output of the IIR averaging filter (step 106), a simple absolute level detector may be used for establishing the signal power's initial minimum and maximum level. Accordingly, the initial minimum and maximum power levels are the same.

[0035] Once the initial minimum and maximum power levels have been determined, they may be tracked, or updated, using a slow first-order averaging filter to follow the signal's dynamic change. ("Slow" in this context means a time constant of seconds, relative to typical gaps and pauses in voice conversation.) Accordingly, the minimum and maximum power levels will begin to diverge. Thus, after several frames, the minimum and maximum power levels will

reflect an accurate measure of the actual minimum and maximum values of the input signal power. In one example, the minimum and maximum power levels are not considered to be sufficiently accurate until the gap between them has surpassed an initial signal level gap. In this particular example, the initial signal level gap is 12dB, but may differ as will be appreciated by one of ordinary skill in the art. Referring to Figure a 4, a sample output of the minimum and maximum signal levels is shown. (See also line 154 in Figure 1.)

[0036] Further, in order to provide a high level of stability for inhibiting the power level gap from collapsing, the slow first-order averaging filter for tracking the minimum power level may be designed such that it is quicker to adapt to a downward change than an upward change. Similarly, the slow first-order averaging filter for tracking the maximum power level may be designed such that it is quicker to adapt to an upward change than a downward change. In the event that the power level gap does collapse, the system may be reset to establish a valid minimum/maximum baseline.

[0037] In step 110, using the slow-adapting minimum and maximum power levels as a baseline, a range of signals are defined as noise and voice respectively. A noise power level threshold is set at minimum power level + x dB, and a voice power level threshold is set at maximum power – y dB. For the purpose of this step, any signals whose power falls below the noise power level threshold are considered noise. A sample noise power profile against the pre-selected frequency components is illustrated in Figure 5. (See also line 156 in Figure 1.) Similarly, any signals whose power falls above the voice power level threshold are considered voice. A sample voice power profile against the frequency components is illustrated in Figure 6. (See also line 158 in Figure 1.) A first-order IIR averaging filter may be used to track the slowly-changing noise power and voice power. It should be noted that the margin values, x and y, used to set the noise and voice threshold need not be the same value.

[0038] In step 111, once the noise power and voice power profiles have been established, a pri-SNR profile against the frequency components of the signal is calculated in accordance with Equation 1. The pri-SNR profile is subsequently tracked on a frame-by-frame basis using a first-order IIR averaging filter having the noise and voice power profiles as its input. Referring to Figure 7, a sample pri-SNR profile is shown. (See also line 160 in Figure 1.)

[0039] In step 112, in parallel with the pri-SNR calculation, as the noise power profile against frequency components becomes available, the post-SNR profile is obtained by dividing each frequency component's instant power against the corresponding noise power, in accordance with Equation 2. In step 113, as both the pri-SNR and post-SNR profiles become available for each signal frame, the LLR value can be calculated in accordance with Equation 3 on a frame-by-frame basis.

[0040] In step 114, the LLR threshold is established by averaging the LLR values corresponding to the signal frames whose power falls within the noise level range established in step 110. The LLR threshold may be subsequently tracked using a first-order IIR averaging filter. As an alternative, once the LLR threshold has been established and VAD decisions are occurring on a frame-by-frame basis, subsequent LLR threshold updating and tracking can be achieved by using the noise LLR values when the VAD output indicates the frame is noise.

[0041] The result is shown in Figures 8 and 9. Referring to Figure 8, a sample of LLR distribution over time is illustrated. (See also line 162 in Figure 1.) Referring to Figure 9, a smaller scale portion of the LLR distribution in Figure 8 is illustrated, with the LLR threshold superimposed. (See also line 164 in Figure 1.) According to the LLR calculations, results at zero and below are likely to be noise. The further below zero the result, the more likely it is to be noise. It should be noted that although some frames may have been considered as noise in the step 110, this determination is not reliable enough for VAD. This fact is illustrated in Figure 9, where some of the LLR values for frames that would have been categorized as noise in step 110 are well above zero.

[0042] In step 116, once the LLR threshold has been established, silence detection is initiated on a frame-by-frame basis. The number of LLR values required before the LLR threshold is considered to be established is implementation dependent. Typically, the greater the number of LLR values required before considering the threshold established, the more reliable the initial threshold. However, more LLR values requires more frames, which increases the response time. Accordingly, each implementation may differ, depending on the requirements and designs for the system in which it is to be implemented. Once the threshold has been established, a frame is considered as silent if its LLR value is below LLR threshold + m dB, where m dB is a predefined margin. Typically, LLR threshold + m dB is below zero with sufficient margin. Further, silence

suppression is not triggered unless there are *h* number of consecutive silence frames, also referred to as a hang-over time. A typical hang over time is 100ms, although this may vary as will be appreciated by a person skilled in the art. Referring to Figure 10, a noise-removed voice signal in accordance with the present embodiment is illustrated. (See also line 166 in Figure 1.)

[0043] It should also be noted that the forgetting factors used in every first-order IIR averaging filter can be individually tuned to achieve optimal overall performance, as will be appreciated by a person of ordinary skill in the art.

[0044] Figure 11 is a block diagram of a communications device 200 implementing an embodiment of the present invention. The communications device 200 includes an input block 202, a processor 204, and a transmitter block 206. The communications device may also include other components such as an output block (e.g., a speaker), a battery or other power source or connection, a receiver block, etc. that need not be discussed in regard to embodiments of the present invention. As an example, the communications device 200 may be a cellular telephone, cordless telephone, or other communications device concerned about spectrum or power efficiency.

[0045] The input block 202 receives input signals. As an example, the input block 202 may include a microphone, an analog to digital converter, and other components.

[0046] The processor 204 controls voice activity detection as described above with reference to Figure 1. The processor 204 may also control other functions of the communication device 200. The processor 204 may be a general processor, an application-specific integrated circuit, or a combination thereof. The processor 204 may execute a control program, software or microcode that implements the method described above with reference to Figure 1. The processor 204 may also interact with other integrated circuit components or processors, either general or application-specific, such as a digital signal processor, a fast Fourier transform processor (see step 102), an infinite impulse response filter processor (see step 106), a memory to store interim and final results of processing, etc.

[0047] The transmitter block 206 transmits the signals resulting from the processing controlled by the processor 204. The components of the transmitter block 206 will vary depending upon the needs of the communications device 200.

[0048] Although the invention has been described with reference to certain specific embodiments, various modifications thereof will be apparent to those skilled in the art without departing from the spirit and scope of the invention as outlined in the claims appended hereto.